(12)

ASI Adaptive Sensors, Incorporated
216 Pico Blvd., Suite 8, Santa Monica, CA 90405 (213) 396 5997

NEW SYNDROME DECODING TECHNIQUES FOR CONVOLUTIONAL CODES OVER GF(q)

I. S. Reed

January 1983

FINAL REPORT

Submitted to

The NAVAL AIR SYSTEMS COMMAND
Washington, D.C. 20361

Contract N00019-81-C-0541

by

ADAPTIVE SENSORS, INC.
216 Pico Boulevard, Suite 8
Santa Monica, CA 90405

CONTENTS

FIGURES

NEW SYNDROME DECODING TECHNIQUES FOR CONVOLUTIONAL CODES OVER GF(q)

I. S. Reed

I. INTRODUCTION

This is a final report on a one-year study of techniques for improving the encoding and decoding of error-correcting codes. For the previous three quarters emphasis was given to methods for making the encoding and decoding of Reed-Solomon codes more efficient. These methods included the possibility [1] of using Winograd's fast transforms for transform encoding and decoding of Reed-Solomon codes, and studies for simplifying the arithmetic of the Galois fields used in error-correcting codes. The latter topic ~~lead~~ led to an investigation [2] of the architecture needed to realize Berlekamp's new bit-serial multiplier [3] of Galois field elements and its generalization [4] to a symbol-serial multiplier. During the last quarter a new syndrome decoding algorithm for convolutional codes (CC) was conceived by the author [5]. In ~~the present~~ this report this idea is extended to a greater generality.

First, the algebraic nature and structure of convolutional codes over a Galois field GF(q) is reviewed and developed. This background is then used to find the general solution of the syndrome equations for the error polynomial vector $e(D)$ where D is the unit delay operator. In particular it is shown that these syndrome equations are linear Diophantine equations over the ring of polynomials in D and with coefficients in GF(q). The methods of solving linear Diophantine equations for the integers are then used to solve the syndrome equations for $e(D)$.

The set of Diophantine solutions of the syndrome equations constitutes a coset of the convolutional code space or subgroup. The problem of syndrome decoding is to find the minimum weight polynomial vector $\hat{e}(D)$ of this coset to subtract from the received polynomial vector $\underline{z}(D)$ to yield an estimate $\hat{y}(D)$ of the transmitted polynomial vector. In order to find $\hat{\underline{e}}(D)$ efficiently a new recursive, Viterbi-like algorithm is devised. This new syndrome decoding algorithm is presented in this report by example.

For a fixed convolutional code the new recursive syndrome decoder for CC appears to be comparable in complexity to the Viterbi decoder except that in the new decoder fewer comparisons are required and the control logic is considerably simpler. However, if one wishes to design a CC decoder for several different rate codes of the same constraint length, it appears that the principles of the new syndrome decoder may yield a simpler system than one could achieve using the Viterbi methods. Thus for variable rate communication systems that utilize convolutional codes the new syndrome decoding concept appears to have an advantage over the standard Viterbi decoding techniques. A precise quantification of this comparison is a topic for future study.

II. ALGEBRAIC STRUCTURE OF CONVOLUTIONAL CODES (CC)

Let a_0, a_1, a_2 ... be any sequence of symbols from the finite field $F = GF(q)$ of q elements. Further let D be the unit delay operator. D operates on a function x(n) of discrete time in accordance with the definition

$$D\,x(n) = x(n-1) \qquad (1)$$

for all n. In terms of D the sequence $\{a_j\}$ is conveniently represented by what is called its D-transform,

$$A(D) = a_0 + a_1 D + a_2 D^2 + \dots , \tag{2}$$

in powers of the operator D.

One way to understand A(D) is to consider A(D) to be the output of a symbol generating box. At the present instant of time the output is a_0; at one unit of time later the output is a_1, the coefficient of D; at two units of time later the output is a_2, the coefficient of D^2; and so forth. Looked upon in this manner the D-transform converts an abstract sequence $\{a_j\}$ of symbols from GF(q) into the same sequence of symbols but now ordered in time. Assume in (2) that the coefficients commute with D^j.

The input to a convolutional encoder is a set of k discrete-time input sequences. In terms of D-transforms this input is represented by the vector

$$x(D) = [x_1(D), x_2(D), \dots , x_k(D)] \tag{3}$$

where

$$x_j(D) = x_{0j} + x_{1j} D + x_{2j} D^2 + \dots$$

for $(j = 1, 2, \dots k)$ with coefficients in GF(q).

Very simply an encoder for a CC is some <u>linear</u> sequential circuit over the finite field GF(q) with vector input x(D) and vector output

$$y(D) = [y_1(D), y_2(D), \dots , y_n(D)] \tag{4}$$

where n > K and

$$y_r(D) = y_{0r} + y_{1r} D + y_{2r} D^2 + \dots$$

for (r = 1, 2, ... n). For the standard (n, k) convolutional code ((n, k) CC) the linear relationship between the input and output is assumed to have finite memory so that it can be expressed as a matrix convolution of form

$$y(D) = x(D)\, G(D) \tag{5}$$

where G(D) is a K x n matrix of polynomials of finite degree in D over GF(q). G(D) in (5) is usually called the generating matrix of the (n, k) CC or k/n rate CC and has the specific form, as a k x n matrix,

$$G(D) = \begin{bmatrix} g_{11}(D), & g_{12}(D), & \dots & g_{1n} \\ g_{21}(D), & g_{22}(D), & \dots & g_{2n} \\ \vdots & & & \\ g_{k1}(D), & g_{k2}(D), & \dots & g_{kn} \end{bmatrix} \tag{6}$$

The maximum degree M of the polynomials in G(D) is called the memory, and the constraint length of the code is L = M + 1.

The elements of G(D) in (G) are polynomials in D over the finite field F = GF(q). The set of all such polynomials in D over a field F is an infinite ring F[D] as well as an integral domain since it has no divisors of zero. It can also be demonstrated that F[D] is a Euclidean ring (see [6, Secs. 3.7 and 3.9).

If the elements of G(D) in (6) had been restricted only to members of field F, it would generate a one dimensional vector space over field F. However, since G(D) is a k x n matrix with elements in F[D], the integral domain in D over F, G(D) generates what is called a module over the ring F[D]. A module has the same postulates as a vector space except that its scalars are elements of a ring rather than a field.

To characterize the algebraic properties of different generating matrices G(D) over integral domain F[D] we follow the lead of Forney in [7]. Forney bases his study of CC on the well-known invariant-factor theorem of matrices over an integral domain. The statement of this theorem is reproduced for ring F[D] as follows:

Invariant-Factor Theorem: *If F[D] is the integral domain F[D] and* G is a k x n matrix over F[D], then G has the invariant-factor decomposition

$$G = A \Gamma B \tag{7}$$

where A is a k x k matrix over F[D] with an inverse A^{-1} with elements in F[D]; B is a n x n matrix over F[D] also with an inverse B^{-1} with elements in F[D]; and Γ is a k x n matrix over F[D] of form

$$\Gamma = [\Gamma_1, 0] \tag{8}$$

with 0, a k x n - k matrix of zeros and Γ_1 a diagonal matrix of form

$$\Gamma_1 = \begin{bmatrix} \gamma_1, & 0 & \dots & 0 \\ 0, & \gamma_2 & \dots & 0 \\ \vdots & & & \\ 0, & 0, & \dots & \gamma_k \end{bmatrix} \qquad (9)$$

over R.

The diagonal elements γ_i in (9) for $1 \leq i \leq k$ are elements of F[D] and are called the invariant factors of G. The invariant factors are unique and can be computed as follows: Let $\Delta_0 = 1$. Let Δ_i be the greatest common divisor (GCD) of all i x i subdeterminants (minors) of G. Then $\gamma_i = \Delta_i/\Delta_{i-1}$. If $\gamma_{i+1} \neq 0$, then γ_i divides γ_{i+1} for i = 1, 2, ... k - 1.

Forney in [6] sketches a proof of this theorem. Other more elementary and detailed expositions of this theorem can be found in certain classic works on modern algebra, e.g., see [8; Sec. 10, Ch. III]. Rather than give a proof of this theorem it is perhaps better here to illustrate the computational technique involved by an example.

For this example let F = GF(2) and consider the generating matrix,

$$G(D) = G = \begin{bmatrix} 1 & 1+D & 1+D \\ 1+D & D & 0 \end{bmatrix} = A \begin{bmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \end{bmatrix} B \qquad (10)$$

where A is a 2 x 2 matrix and B is a 3 x 3 matrix, both over F[D]. Matrices A and B can be obtained as a product of elementary row and column operations, respectively. The elementary operations are of three types:

i) The interchange of any two rows (columns) is an elementary operation of type 1.

ii) Let any row (column) be multiplied by an element of F[D]. The addition of this result to any other row (column) is an elementary operation of type 2.

iii) The multiplication of any row (column) by a nonzero scalar of F[D], i.e., a nonzero element of F, is an elementary operation of type 3.

One procedure for finding A and B in (10) is to express (10) in the form,

$$\begin{bmatrix} 1 & 1+D & 1+D \\ 1+D & D & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} G \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

and reduce the left side to the form of Γ in (10) by elementary transformation. To put zeros in the second column of the first row multiply the first column by 1 + D and add the result to the second column of the matrix on the left. This same transformation is performed at the same time on the 3 x 3 identity matrix on the right side of the equation. The result is

$$\begin{bmatrix} 1, & 0, & 1+D \\ 1+D, & 1+D+D^2, & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} G \begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where on the right a new 3 x 3 identity matrix multiplies the elementary transformation matrix

$$\begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

of type 2. Proceeding step by step in this fashion it is readily verified that the left side of (11) reduces finally to

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} G \begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1+D \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} G \begin{bmatrix} 1 & 1+D & D+D^2 \\ 0 & D & 1+D^2 \\ 0 & 1+D & 1+D+D^2 \end{bmatrix} = A^{-1} G B^{-1} \qquad (12)$$

For F = GF(2) it is easy to verify that an elementary matrix E over F[D] is its own inverse, i.e., $E^{-1} = E$ for an elementary matrix of types 1, 2 or 3. Thus solving for G in (12) yields

$$G(D) = \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & D \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1+D \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1+D & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 1+D & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1+D & 1+D \\ 0 & 1+D+D^2 & 1+D^2 \\ 0 & 1+D & D \end{bmatrix} = A\Gamma B \qquad (13)$$

as the invariant-factor decomposition of G(D) in (10).

Let the generating matrix of a CC have the invariant-factor decomposition in (7). Then the output of the encoder in (5) can be expressed as

$$y(D) = x(D)\ G(D) = x(D)\ A(D)\ \Gamma(D)\ B(D) \qquad (14)$$

where A(D) and B(D) have inverses $A^{-1}(D)$ and $B^{-1}(D)$, respectively, and $\Gamma(D)$ is the k x n matrix

$$\Gamma(D) = \begin{bmatrix} \gamma_1(D) & 0 & \dots & 0 & \dots & 0 \\ 0 & \gamma_2(D) & \dots & 0 & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & \gamma_k(D) & \dots & 0 \end{bmatrix} \qquad (15)$$

of invariant factors $\gamma_1(D), \dots \gamma_k(D)$.

For the encoding operation in (14) to be useful in the context of a communications system it is desirable that the mapping of k-vectors x(D) onto n-vectors y(D) over F[D] be one-to-one and reversible. For this to be true there must exist a right inverse matrix $G^{-1}(D)$ of matrix G(D). If $G^{-1}(D)$ exists, then

$$y(D)\ G^{-1}(D) = x(D)\ G(D)\ G^{-1}(D) = x(D)\ I_k = x(D) \qquad (16)$$

where I_k is the k x k identity matrix and x(D) is uniquely recoverable from the encoded message y(D). It will be assumed henceforth that the generating matrix G(D) has a right inverse and that this inverse is realizable in finite delay time.

By the invariant-factor theorem

$$G(D) = A(D)\ \Gamma(D)\ B(D)$$

Thus for $G^{-1}(D)$ to exist the last invariant factor of $\Gamma(D)$ must not be zero, i.e., $\gamma_k \neq 0$. For otherwise, if $\gamma_k = 0$, then G(D) would have rank less than k and as a consequence no inverse.

If $\gamma_k \neq 0$, then it can be verified that

$$G^{-1}(D) = B^{-1}(D)\ \Gamma^*(D)\ A^{-1}(D) \qquad (17)$$

is an inverse for G(D) where $\Gamma^*(D)$ is an n x k matrix with diagonal elements γ_i^{-1} of form

$$\Gamma^*(D) = \begin{bmatrix} \Gamma_1^{-1}(D) \\ 0 \end{bmatrix} \qquad (18)$$

with $\Gamma_1^{-1}(D)$ the inverse of $\Gamma_1(D)$ in (9), i.e.,

$$\Gamma_1^{-1}(D) = \begin{bmatrix} \gamma_1^{-1}(D), & 0 & \cdots & 0 \\ 0 & \gamma_2^{-1}(D) & \cdots & 0 \\ \vdots & \vdots & & \\ 0 & 0 & \cdots & \gamma_K^{-1}(D) \end{bmatrix} \tag{19}$$

If deg $\gamma_K(D) > 0$, then by (19), (18) and (17) the circuit to realize (16) would not be feedback-free.

Massey and Sain [9] proved that an inverse $G^{-1}(D)$ or some delayed version of it must be feedback-free in order to avoid CC that give rise to catastrophic error propagation. Therefore, for an encoder to be useful one must choose it to be feedback-free. If $G^{-1}(D)$ is feedback-free, deg $\gamma_k(D) = 1$ and $\gamma_1 = \gamma_2 \cdots \gamma_k = 1$ so that Γ must have the form

$$\Gamma = [I_k, 0] \tag{20}$$

where I_K denotes a K x K identity matrix and 0 denotes a K x (n - K) matrix of zeros.

The above properties needed for a useful encoder are recapitulated in the following definition of a basic encoder given by Forney [7].

Definition: A basic encoder G(D) is a CC with a feedback-free inverse $G^{-1}(D)$. Both G(D) and $G^{-1}(D)$ are polynomial matrices over F[D], such that $G(D)\ G^{-1}(D) = I_k$.

It was shown in detail by Forney [7] and briefly above that an encoder is basic if and only if it is polynomial over F[D] and has all its invariant factors equal to one. Henceforth in this report only basic encoders for CC will be treated so that by (20) and the invariant-factor theorem the generating matrix for an (n, k) CC has the form

$$G(D) = A(D)\ [I_k, 0]\ B(D) \tag{21}$$

where I_k is the k x k identity matrix.

Forney in [7, Appendix I] exhibits a parity-check matrix H(D) over F[D] for a generating matrix which is equivalent to G(D) in (21). It is shown here that this parity-check matrix is, in fact, the parity-check matrix of all generating matrices equivalent to (21) in the sense of Forney.

To treat the parity-check matrix the Euclidean ring F[D] is extended to the field F(D) of quotients or rational functions of polynomials in F[D], e.g., see [8; Sec. 3.8]. In terms of sequences field F(D) is in one-to-one correspondence with the field S of all possible infinite sequences that can be generated by an impulse passed through all finite memory linear circuits with or without feedback.

Let $g^{(1)}(D)$, $g^{(2)}(D)$, ..., $g^{(k)}(D)$ be the k rows of matrix G(D) in (21). Since G(D) has rank k,

$$V = \left\{ \sum_{i=1}^{k} \alpha_j(D)\ g^{(i)}(D) \,\middle|\, \alpha_j(D) \in F(D) \right\}$$

is a k-dimensional vector space with respect to the field F(D) of scalars or it is equivalent to the field S of sequences. The null space $V^{\perp}$ of V is a vector space with field F(D) of scalars of dimension n - k (see [9, Sec. 3.2]). Let H(D) be any matrix with coefficients in ring F[D] of rank $n - \bar{k}$ which has its row space equal to $V^{\perp}$. Then the rows of H(D) constitute a basis for $V^{\perp}$. Thus V is a null space of $V^{\perp}$ if and only if for any vector $y(D) \varepsilon V$,

$$y(D)\ H^T(D) = 0 \qquad (22)$$

Since $g^{(j)}(D) \varepsilon V$ for $(j = 1, 2, \ldots, \bar{k})$ condition (22) implies,

$$G(D)\ H^T(D) = 0 \qquad (23)$$

An explicit method for constructing a parity matrix H(D) of all generating matrices G(D) with form (21) will now be given. The coefficients of the parity-check matrix H(D), developed by this technique, will be polynomials in D, i.e., elements from the Euclidean ring F[D].

Let the first k rows of matrix G in (21) be a submatrix B_1 and the last (n - k) rows of B be a submatrix B_2. Then B is the matrix

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \qquad (24)$$

in terms of submatrices B_1 and B_2. Similarly denote the first k columns of the inverse matrix B^{-1} of B by $B_1^{-\prime}$ and denote the last (n - k) columns of B^{-1} by B_2'. Then the inverse of B is the matrix

$$B^{-1} = [B_1', B_2'] \qquad (25)$$

Multiplying (24) and (25) yields

$$BB^{-1} = \begin{bmatrix} B_1B_1', & B_1B_2' \\ B_2B_1', & B_2B_2' \end{bmatrix} = \begin{bmatrix} I_k, & 0 \\ 0, & I_{n-k} \end{bmatrix} \qquad (26)$$

and the matrix identities,

$$\begin{aligned} &B_1B_1' = I_K, \quad B_2B_1' = 0 \quad \text{and} \\ &B_2B_1' = 0, \quad B_2B_2' = I_{n-k} \end{aligned} \qquad (27)$$

Let

$$H(D) = (B_2')^T, \qquad (28)$$

where "T" denotes transpose and B_2' is defined in (25), be a candidate for the parity-check matrix of G(D) in (21). Multiplying G(D) in (21) by the transpose of H(D) in (28) produces by (24) and (27),

$$\begin{aligned} G(D)\ H^T(D) &= A[I_k, 0] \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} B_2' \\ &= [A, 0] \begin{bmatrix} B_1B_2' \\ B_2B_2' \end{bmatrix} \\ &= [A, 0] \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} = 0 \end{aligned} \qquad (29)$$

Since the n - k rows of H(D) are the n - k columns of the invertible matrix B^{-1}, these rows must be linearly independent. Thus H(D), as defined in (2P), has rank (n - k) and is a valid parity-check matrix for the general basic encoder G(D), given in (21).

The parity-check matrix H(D), associated with the general k x n generator matrix G(D) of a basic encoder, is given by (28). However, for two special cases, the systematic (n, k) CC and the non-systematic (n, 1) CC a parity-check matrix can be found more directly. For example, for the systematic (n, k) CC the parity-check matrix has form G(D) = $[I_k, P(D)]$ where I_k is the k x k identity matrix and P(D) is a k x (n - k) matrix of polynomials over F = GF(q). In this case it is readily verified [9] that $H(D) = [-P^T(D), I_{n-k}]$ is a parity-check matrix.

The generator matrix for the non-systematic (n, 1) CC has by (6) the form

$$G(D) = [g_1(D), g_2(D), \dots g_n(D)] \qquad (30)$$

where for simplicity the first subscript of $g_{1j}(D)$ has been dropped. Condition (23) for the (1 x n) generating matrix G(D) in (30) is easily shown to be satisfied by the (n - 1) x n matrix

$$H(D) = \begin{bmatrix} g_2(D), & g_1(D), & 0, & \dots, & 0 \\ g_3(D), & 0 & g_2(D), & \dots, & 0 \\ \vdots & & & & \\ g_n(D), & 0, & 0, & \dots, & G_1(D) \end{bmatrix} \qquad (31)$$

so that it is a parity-check matrix of the (n, 1) CC generated by G(D) in (30).

To illustrate how to use (28) to compute a parity-check matrix consider again the example of a generating matrix, given in (10). By (12) the matrix B^{-1} for this G is

$$B^{-1} = \begin{bmatrix} 1 & 1+D & D+D^2 \\ 0 & D & 1+D^2 \\ 0 & 1+D & 1+D+D^2 \end{bmatrix} \qquad (32)$$

so that by (25) and (28)

$$H(D) = (B_2')^T = [D + D^2, 1 + D^2, 1 + D + D^2], \qquad (33)$$

the transpose of the last column of B^{-1} in (22). It is easily verified that (23) satisfies (23), i.e., $G(D)\ H^T(D) = 0$, and that H(D) is of rank one. Hence H(D) in (23) is the parity-check matrix of the (3, 2) CC with generating matrix G(D) in (10).

In the next section the parity-check matrix will be used to obtain the syndrome of the received CC. A new decoding algorithm will then be presented by example.

III. SOLUTIONS OF SYNDROME EQUATION FOR CONVOLUTIONAL CODES

Let y(D) be transmitted CC in accordance with (5) where G(D) is the generating matrix for a basic encoder. Next let $z(D) = y(D) + e(D)$ be the received code possibly corrupted by an error or noise sequence e(D). The syndrome S(D) of z(D) is defined by

$$S(D) = z(D)\ H^T(D) \qquad (34)$$

where H(D) is syndrome (28) for the basic encoder.

Substituting (5) in syndrome (34) yields

$$S(D) = (y(D) + e(D))\ H^T(D) = e(D)\ H^T(D) + x(D)\ G(D)\ H^T(D) \tag{35}$$

But by construction (23) is satisfied, i.e., $G(D)\ H^T(D) = 0$. Hence the last term of (35) vanishes and

$$S(D) = e(D)\ H^T(D) \tag{36}$$

is the syndrome in terms of an error sequence or polynomial e(D). The syndrome for a basic (n, k) CC is by (36) totally independent of the transmitted coded message y(D). The syndromes of block group codes also have this property so one might suspect that it would be possible to use syndromes to decode CC in a manner similar to that used for block codes.

The first step towards achieving this goal for CC, analogous to that used for block codes, is to find the general solution for e(D) of the syndrome equation (36), assuming that S(D) is computed by (34). That is, given S(D) by (34), solve for the set of all solutions e(D) of the syndrome equation (36).

To find the general solution of (36) again use is made of the important invariant-factor theorem of the last section. This theorem is applied to the matrix

$$M(D) = H^T(D) \tag{37}$$

the transpose of the parity-check matrix in (36). By construction the rank of H(D) is n - k, the maximum possible rank. Hence M(D) has the

Smith* normal form of the invariant-factor theorem,

$$M(D) = L(D) \begin{bmatrix} \Lambda \\ 0 \end{bmatrix} R(D) = H^T(D) \tag{38}$$

where L(D) and R(D) are invertible n x n and (n - k) x (n - k) matrices over F(D), $\Lambda = \text{diag}\ (\lambda_1, \lambda_2, \ldots \lambda_{n-k})$

$$\Lambda = \text{diag}\ (\lambda_1, \lambda_2, \ldots \lambda_{n-k}) \tag{39}$$

and "0" denotes a k x (n - k) matrix of zeros. The λ_j's in (39) are the invariant factors defined as follows: Let $\delta_0 = 1$, let δ_j be the GCD of all j x j minors of M. Then $\lambda_j = \delta_i/\delta_{i-1}$ and λ_j divides λ_{j+1} for j = 1, 2, ..., n - k - 1.

A lemma, due to Forney [7, Appendix I], is used to evaluate the diagonal matrix Λ of invariant factors in (38). In the present terminology this lemma is requoted as follows:

Lemma (Forney): The (n - k) x (n - k) minors of H(D) are equal up to scalar field elements in F = GF(q) to the k x k minors of G(D).

Since the basic encoder has a generating matrix G(D) of form (21), the k x k minors of G(D) have a GCD equal to 1. Hence by Forney's lemma the (n - k) x (n - k) minors of H(D) also have a GCD equal to 1. Thus

$$1 = \delta_{n-k} = \delta_{n-k-1} = \cdots = \delta_1$$

*The invariant-factor theorem was developed by the British mathematician H.J.S. Smith in the middle of the 19th century.

and the invariant factors of $H^T(D)$ are all 1. Therefore, $\Lambda = I_{n-k}$ and by (38)

$$M(D) = L(D) \begin{bmatrix} I_{n-k} \\ 0 \end{bmatrix} R(D) = H^T(D) \qquad (40)$$

is the Smith normal form of the transpose of a parity-check matrix for a basic encoder.

To solve for e(D) first substitute expression (40) for $H^T(D)$ into (36). This yields

$$S(D) = e(D)\ L(D) \begin{bmatrix} I_{n-k} \\ 0 \end{bmatrix} R(D) \qquad (41)$$

Next multiply both sides of (41) by $R^{-1}(D)$ to obtain

$$\sigma(D) = S(D)\ R^{-1}(D) = [e(D)\ L(D)] \begin{bmatrix} I_{n-k} \\ 0 \end{bmatrix} = \varepsilon(D) \begin{bmatrix} I_{n-k} \\ 0 \end{bmatrix} \qquad (42)$$

where

$$\sigma(D) = S(D)\ R^{-1}(D) = [\sigma_1(D), \ldots, \sigma_{n-k}(D)] \qquad (43)$$

is an (n - k)-component transformed vector of S(D) and

$$\varepsilon(D) = e(D)\ L(D) = [\varepsilon_1(D), \ldots, \varepsilon_n(D)] \qquad (44)$$

is an n-component transformed vector of the unknown polynomial error vector e(D).

The component-by-component solution of (42) is obtained by equating comoonents of the equation

$$\varepsilon_j(D) = \sigma_j(D) \qquad \text{for} \qquad (j = 1, 2, \ldots, n - k) \tag{45a}$$

and

$$\varepsilon_j(D) = t_{j-n+k}(D) \qquad \text{for} \qquad (j = n - k + 1, \ldots, n) \tag{45b}$$

where $t_i(D)$ for $(i = 1, 2, \ldots, k)$ is an arbitrary polynomial in the Euclidean ring $F[D]$. Substituting (45a) and (45b) into the right side of (44) and solving for $e(D)$ yields finally

$$e(D) = [e_1(D), \ldots, e_n(D)]$$

$$= [\sigma_1(D), \ldots, \sigma_{n-k}(D), t_1(D), \ldots, t_k(D)]\ L^{-1}(D) \tag{46}$$

as the general solution of the syndrome equation (36) in terms of the $n - k$ components of the transformed syndrome $\sigma(D)$ in (42) and k arbitrary polynomials parameters $t_j(D)$ of $F[D]$ for $(j = 1, 2, \ldots, k)$.

Some examples of the above technique for solving the linear Diophantine equations of the syndrome equation are now presented. Consider first the generating matrix

$$G(D) = [1 + D^2, 1 + D + D^2], \tag{47}$$

of a (2, 1) CC of constraint length $L = 3$. It is easily deomonstrated that

$$H^T(D) = \begin{bmatrix} 1+D+D^2 \\ 1+D^2 \end{bmatrix}$$

is the transpose of parity-check matrix. Diagonalizing $H^T(D)$ with elementary transformations yields

$$H^T(D) = \begin{bmatrix} 1+D+D^2, & 1+D \\ 1+D^2, & D \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = L(D) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad (48)$$

as the Smith normal form for $H^T(D)$. Substituting (48) in (36) produces

$$S(D) = [e_1(D), e_2(D)] \begin{bmatrix} 1+D+D^2, & 1+D \\ 1+D^2, & D \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= [(1 + D + D^2)\, e_1(D) + (1 + D^2)\, e_2(D),\ (1 + D)\, e_1(D) + e_2(D)] \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= [\varepsilon_1(D), \varepsilon_n(D)] \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad (49)$$

The general solution of this equation for ε_1 and ε_2 is

$$\varepsilon_1(D) = S(D) \quad \text{and} \quad \varepsilon_2(D) = t(D) \qquad (50)$$

where $t(D)$ is an arbitrary polynomial. The linear relation $\varepsilon(D) = e(D)\ L(D)$ is solved by

$$e(D) = [e_1(D), e_2(D)] = \varepsilon(D)\ L^{-1}(D)$$

$$= [e_1(D), e_2(D)] \begin{bmatrix} D, & 1+D \\ 1+D^2, & 1+D+D^2 \end{bmatrix}$$

$$= [S(D), t(D)] = \begin{bmatrix} D & 1+D \\ 1+D^2, & 1+D+D^2 \end{bmatrix}$$

$$= [DS(D) + (1 + D^2)\ t(D), (1 + D)\ S(D) + (1 + D + D^2)\ t(D)]$$

where the solutions S(D) and t(D) in (49) have been substituted for $\varepsilon_1(D)$ and $\varepsilon_2(D)$, respectively. Equating coefficients yields

$$e_1(D) = DS(D) + (1 + D^2) + (D)$$

$$e_2(D) = (1 + D)\ S(D) + (1 + D + D^2)\ t(D) \qquad (51)$$

as the general solution for the syndrome equation in (36) for this example of a CC where t(D) is an arbitrary element in F[D]. That (51) is,in fact,the solution of (36) can be verified by substitution.

For the next example consider the generating matrix

$$G(D) = [1 + D^2, 1 + D + D^2, 1 + D + D^2] \qquad (52)$$

of a (3, 1) CC with constraint length 3. Using the Forney method developed in the last section, a parity-check matrix for G(D) is readily calculated to be

$$H(D) = \begin{bmatrix} 1+D+D^2, & 1+D^2, & 0 \\ 1+D+D^2, & D^2, & 1 \end{bmatrix} \tag{53}$$

The Smith normal form for $H^T(D)$ is given by

$$H^T(D) = L \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{54}$$

where the inverse of L is

$$L^{-1} = \begin{bmatrix} D & 1+D & D \\ 0 & 0 & 1 \\ 1+D^2 & 1+D+D^2 & 1+D+D^2 \end{bmatrix} \tag{55}$$

Substituting (53) in (34) the syndrome is $S(D) = z(D)\ H^T(D)$. Hence by (36) and (54) the syndrome equation to solve is

$$S(D) = [S_1(D), S_2(D)] = \left([e_1(D), e_2(D), e_3(D)]\ L\right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$= [\varepsilon_1(D), \varepsilon_2(D), \varepsilon_3(D)] \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{56}$$

By (45a, b) the solution of (56) for $\varepsilon_j(D)$ is

$$\varepsilon_1(D) = S_1(D),\ \varepsilon_2(D) = S_2(D),\ \varepsilon_3(D) = t(D) \qquad (57)$$

where t(D) is an arbitrary element of F[D]. Finally the $e_i(D)$ in terms of the solutions (57) is, using (55),

$$\begin{aligned}[e_1(D),\ e_2(D),\ e_3(D)] &= [S_1(D),\ S_2(D),\ t(D)]\ L^{-1} \\ &= [DS_1(D) + (1 + D^2)\ t(D),\ (1 + D)\ S_1(D) \\ &\quad + (1 + D + D^2)\ t(D),\ DS_1(D) + S_2(D) \\ &\quad + (1 + D + D^2)\ t(D)]\end{aligned}$$

This yields

$$\begin{aligned}e_1(D) &= DS_1(D) + (1 + D^2)\ t(D) \\ e_2(D) &= (1 + D)\ S_1(D) + (1 + D + D^2)\ t(D) \\ e_3(D) &= DS_1(D) + S_2(D) + (1 + D + D^2)\ t(D) \qquad (58)\end{aligned}$$

as the general solution of the syndrome equation (34) for the (3, 1) encoder in (47).

For a final example consider the parity-check matrix, $H(D) = [D + D^2,\ 1 + D^2,\ 1 + D + D^2]$, found in (33) for the (3, 2) CC with generating matrix (10). A diagonization of $H^T(D)$ yields for this example

$$L^{-1}(D) = \begin{bmatrix} 0 & 1+D & D \\ 1 & D+D^3 & D^2+D^3 \\ 0 & 1+D+D^2 & 1+D^2 \end{bmatrix} \qquad (59)$$

where L(D) is left invertible matrix of the Smith normal form in (40) for $H^T(D)$. In this case the syndrome equation to solve for e(D) is

$$S(D) = [e_1(D), e_2(D), e_3(D)]\ L \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= [\varepsilon_1(D), \varepsilon_2(D), \varepsilon_3(D)] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad (60)$$

The solution in (60) for $\varepsilon(D)$ is

$$\varepsilon_1(D) = S(D),\ \varepsilon_2(D) = t_1(D),\ \varepsilon_3 = t_2(D) \qquad (61)$$

where $t_1(D)$ and $t_2(D)$ are arbitrary elements or parameters of F[D]. Solving e(D) L = $\varepsilon(D)$ for e(D) in terms of the solution (61) for $\varepsilon(D)$, is, using (59),

$$[e_1(D), e_2(D), e_3(D)] = [S(D)_1, t_1(D), t_2(D)] \cdot \begin{bmatrix} 0 & 1+D & D \\ 1 & D+D^3 & D^2+D^3 \\ 0 & 1+D+D^2 & 1+D^2 \end{bmatrix}$$

which yields, upon an equating of coefficients,

$$e_1(D) = t_1(D)$$

$$e_2(D) = (1 + D)\ S(D) + (D + D^3)\ t_1(D) + (1 + D + D^2)\ t^2(D)$$

$$e_3(D) = DS(D) + (D^2 + D^3)\ t_1(D) + (1 + D^2)\ t_2(D) \tag{62}$$

as *the general solution of the syndrome equation* (34) *for the* (3, 2) encoder in (10) in terms of two arbitrary parameters $t_1(D)$ and $t_2(D)$ in F[D]. It is a straightforward exercise to verify that (62) satisfies (34). It will be shown in the next section by an example how to use the solutions of the syndrome equation to perform optimum syndrome decoding.

IV. SYNDROME DECODING OF (n, k) CC

Syndrome decoding of an (n, k) CC involves finding a maximum likelihood estimate (MLE) $\hat{e}(D)$ of the actual error sequence in the coset, determined by (46), of all possible solutions of the syndrome equation (34). In order to accomplish this both the weight or distance between codewords of a sequence and the type of channel need to be defined. For an (n, k) CC a possible error sequence is of form $e(D) = [e_1(D), e_2(D), \ldots, e_n(D)]$ where $e_j(D)$ for ($j = 1, 2, \ldots, n$) are finite degree polynomials over GF(q). The usual weight for a discretized channel is the Hamming weight. The Hamming weight of e(D) is

$$W_H[e(D)] = \sum_{j=1}^{n} W_H[e_k(D)] \tag{63}$$

where $W_H[e_j(D)]$, the Hamming weight of $e_j(D)$, is the number of nonzero coefficients of $e_k(D)$. It is convenient for this weight to assume that the channel can be approximated by a q-ary channel (see [1, Sec. 7.2]).

If in (3) deg $[x_j(D)] \leq L - 1$ for $1 \leq j \leq k$, codeword $y(D) = (y_1(D), y_2(D), \ldots, y_n(D))$ is said to be the L-th truncation of an (n, k) CC (see [11, p. 203]). In this case

$$\deg\, [y_i(D)] \leq M + L - 1 \qquad \text{for } 1 \leq i \leq n \tag{64}$$

where M is the memory. Hence an L truncated (n, k) CC can be considered to be a block code where each word has length n(L + M). Hence for a truncated (n, k) CC the MLE of an error vector is what it would be for a linear block code. For a truncated (n, k) CC transmitted over a q-ary symmetric channel the MLE of e(D) is any vector $\hat{e}(D)$ of form (46) such that

$$W_H[\hat{e}(D)] = \underset{t_1(D), \ldots t_k(D)}{\text{Min}} \left(W_H \left| [\, \sigma_1(D), \ldots, \sigma_{n-k}(D), t_1(D), \ldots, t_k(D)]\, L^{-1}(D) \right| \right) \tag{65}$$

The above procedure for finding the MLE $\hat{e}(D)$ or the error vector, needed to correct a codeword, is equivalent to the usual technique for correcting block codes, e.g., see [10, Sec. 7.5]. A recursive technique is developed now by example to perform the minimization required in (65). The iterative minimization procedure, needed to efficiently find $\hat{e}(D)$, is a Viterbi-like or dynamic programming type of algorithm.

As an example of the new syndrome decoding algorithm consider the (3, 1) CC with the generating matrix in (52) If (52) is substituted in

(4), then

$$[y_1, y_2, y_3] = [x + D^2x, x + Dx + D^2x, x + Dx + D^2x] \tag{66}$$

is the output of the encoder. Assume the input sequence is $x = [0\ 1\ 0\ 0\ 1\ 0]$. Then by (66)

$$y_1 = [0\ 1\ 0\ 1\ 1\ 0\ 1\ 0],$$

$$y_2 = [0\ 1\ 1\ 1\ 1\ 1\ 1\ 0],$$

$$y_3 = [0\ 1\ 1\ 1\ 1\ 1\ 1\ 0]$$

are the three components of the transmitted sequence. Let the corresponding three received sequences be

$$z_1 = [1\ 1\ 0\ 1\ 1\ 0\ 1\ 0],$$

$$z_2 = [0\ 1\ 0\ 1\ 1\ 1\ 1\ 0],$$

$$z_3 = [0\ 1\ 1\ 0\ 1\ 1\ 1\ 0] \tag{67}$$

Next substitute the parity-check matrix (53) of the (3, 1) CC in (34) to obtain

$$S_1 = (1 + D + D^2)\ z_1 + (1 + D^2)\ z_2$$

$$S_2 = (1 + D + D^2)\ z_1 + D^2\ z_2 + z_3 \tag{68}$$

as the syndromes of the code. A calculation of the syndromes in (68) in terms of the received sequence $[z_1, z_2, z_3]$ in (67) yields the syndrome sequences

$$S_1 = [1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$$

$$S_2 = [1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0] \qquad (69)$$

for this example. Given the syndrome sequences (69) it is desired now to solve syndrome equation (36) for the error vector sequence e(D).

The explicit general solutions of the syndrome equation in (36) for the components of e(D) were found in (58) of the last section. These solutions are explicitly

$$e_1 = DS_1 + t + Dt$$

$$e_2 = S_1 + DS_1 + t + Dt + D^2t$$

$$e_3 = DS_1 + S_2 + t + Dt + D^2t \qquad (70)$$

where t is an arbitrary polynomial in F[D] and where for simplicity in notation the functional dependence on D of such functions as $S_1(D)$, t(D), etc., is deleted. Note that physically the functions DS_1, D^2t, etc., in (69) can be interpreted as the function $S_1(D)$, delayed by one time unit, and the function t(D), delayed by two time units, respectively.

The problem now is to find from (70) and the given syndrome sequences (69) the MLE $\hat{e}$ of $e = [e_1, e_2, e_3]$. As in the Viterbi decoding algorithm $\hat{e}$ is found iteratively or sequentially with the aid of a trellis diagram (see [12]). However, in the present case the underlying trellis diagram is "universal" in the sense that it is identical for all (n, k) CC of fixed k and constraint length L.

For the present example the states of the trellis diagram are equivalent to the states of the shift register needed in (70) to store sequentially the delayed versions $Dt(D)$ and $D^2t(D)$ of the arbitrary function t(D) on F. The block diagram of a shift register to hold Dt and D^2t, the function t, delayed by one time unit and two time units, respectively, is shown in Figure 1. The state table of the shift register in Fig. 1, when conceived to be a sequential circuit, is given in Figure 3. Finally, in Figure 3 the trellis diagram of the state table in Fig. 2 is presented. A solid-line transition in Fig. 3 corresponds to the input t(D) = 0; a dashed-line transition corresponds to the input t(D) = 1.

The new Viterbi-like syndrome decoding algorithm is illustrated by example in Figure 4. The digits of the syndrome sequences S_1 and S_2 computed in (69) are placed immediately over the corresponding transition paths of the trellis. The vectors $[e_1, e_2, e_3]$ are computed at each stage from equations (70), using the syndrome data for S_1, DS_1 and S_2 and data t, Dt and D^2t, depending on the particular path taken.

To illustrate the above procedure suppose that the algorithm has reached stage 2 at state d = 11. At stage 2 the required values of the syndrome sequence needed in (70) are

$$S_1 = 0, \; DS_1 = 1 \quad \text{and } S_2 = 1 \tag{71}$$

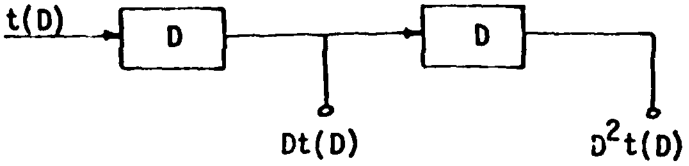


Figure 1. Shift register to generate delayed versions of t(D)

	Dt	D^2t	t = 0	t = 1
a =	0	0	0 0	1 0
b =	0	1	0 0	1 0
c =	1	0	0 1	1 1
d =	1	1	0 1	1 1

Figure 2. State table of shift register for t(D)

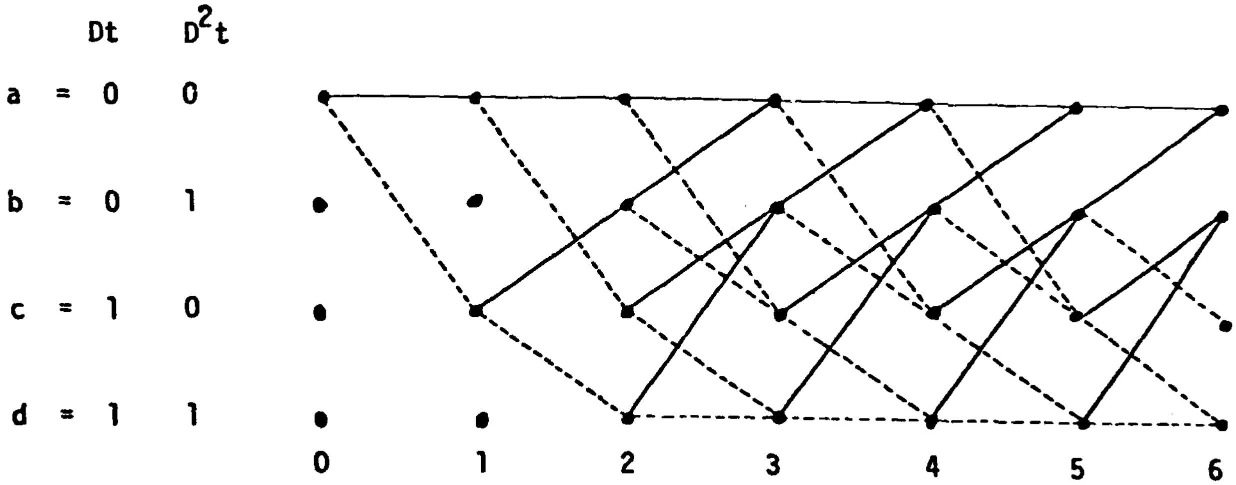
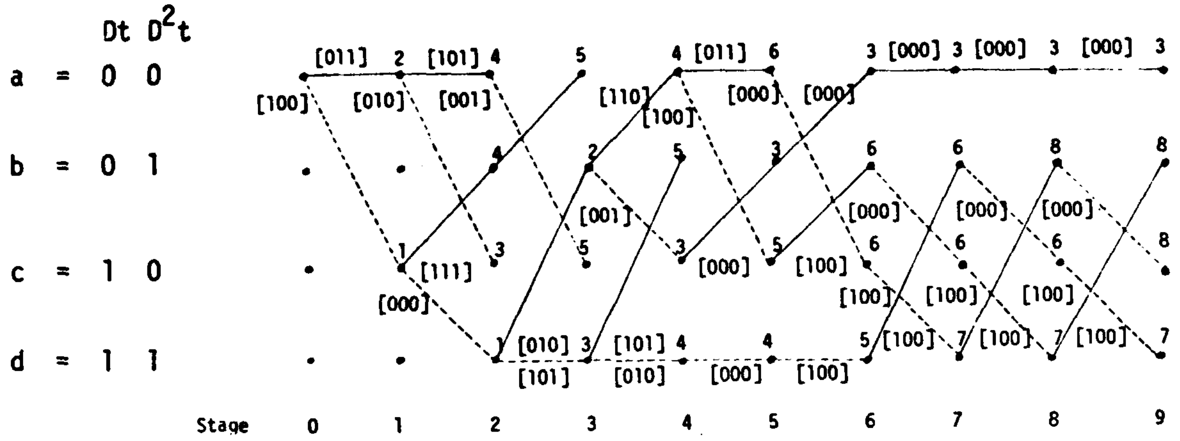


Figure 3. Trellis diagram of shift register for t(D). Input t(D) = 0 is represented by solid line. t(D) = 1 is represented by a dashed line.

$$S_1 = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]$$
$$S_2 = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]$$



$$\hat{e} = [100\ 000\ 010\ 001\ 000\ 000\ 000\ 000\ 000]$$

Note: Each branch of the trellis is labeled with $[e_1, e_2, e_3]$ where $e_1 = DS_1 + t + Dt$, $e_2 = S_1 + DS_1 + t = Dt + D^2t$, and $e_3 = DS_1 + S_2 + t + Dt + D^2t$. Each node at stage k is labeled with $W_H[e_1^{(k)}(D), e_2^{(k)}(D), e_3^{(k)}(D)]$ where $e_j^{(k)}(D)$ is polynomial $e_j(D)$ truncated at degree k.

Figure 4. A new Viterbi-like syndrome decoding algorithm for (3, 1) CC

Since the previous two values of t leading to state d at stage 2 are 1,

$$Dt = 1 \quad \text{and} \quad D^2t = 1 \qquad (72)$$

If the t = 1 branch is taken in the trellis from stage 2 and state d, a substitution of (71) (72) and t = 1 into (70) yields

$$[e_1, e_2, e_3] = [0, 1, 0]$$

as the values of e along that segment of the path.

After stage 2 in Fig. 3 there are always two possible transitions leading to a given node in the trellis. The transition chosen is the one of minimum weight. This is precisely the technique of dynamic programming to determine a minimum weight path. A similar method is used in the Viterbi-algorithm to find a transmitted codeword that is closest in Hamming weight to the received codeword.

The trellis diagram shown in Fig. 4 is completed by the above illustration and the dynamic programming rules for choosing the "survivor" segment of path. At state 9 the minimum weight path in the trellis diagram of Figure 4 is clearly a c d b c b a a a a. The branches of this path yield

$$\hat{e}(D) = [\hat{e}_1(D), \hat{e}_2(D), \hat{e}_3(D)] = [1, D^2, D^3]$$

as the estimate of error vector e(D). Subtracting these estimates of the error from the components of z in (67) produces

$$\hat{y}_1 = [0\ 1\ 0\ 1\ 1\ 0\ 1\ 0]$$

$$\hat{y}_2 = [0\ 1\ 1\ 1\ 1\ 1\ 1\ 0]$$

$$\hat{y}_3 = [0\ 1\ 1\ 1\ 1\ 1\ 1\ 0] \tag{73}$$

The Smith normal form of the (3, 1) CC generating matrix G(D) in (52) is

$$G(D)\ [1\ 0\ 0]\ B$$

where

$$B^{-1} = \begin{bmatrix} 1+D & 1+D+D^2 & 1+D+D^2 \\ D & 1+D^2 & D^2 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus the estimate $\hat{x}(D)$ of message in terms of the estimate of the transmitted codeword $\hat{y}(D)$ is

$$\hat{y} = \hat{x}\ G = \hat{x}[1\ 0\ 0]\ B$$

Solving this relationship for $\hat{x}$,

$$\hat{x} = \hat{y}\ G^{-1} = \hat{y}\ B^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= [\hat{y}_1,\ \hat{y}_2,\ \hat{y}_3] \begin{bmatrix} 1+D & 1+D+D^2 & 1+D+D^2 \\ D & 1+D^2 & 1+D^2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= (1 + D)\ \hat{y}_1 + D\ \hat{y}_2 \qquad (74)$$

since by (17)

$$G^{-1} = B^{-1}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

is the right inverse of G. A substitution of the estimates of $\hat{y}_1$ and $\hat{y}_2$ in (73) into (74) yields finally $\hat{x} = [0\ 1\ 0\ 0\ 1\ 0]$ as the estimate of the original message.

In the above example the new syndrome decoding algorithm produced the original message. However, if the number of errors exceeds the free distance d_f within any interval less than some multiple of the constant length, there may exist two or more paths of the same minimum error weight. In such a case a decoding failure and an erasure should be declared.

V. CONCLUSIONS

In this report four new developments in (n, k) convolutional codes with basic encoders are made. A method of Forney is extended to find the syndrome H(D) of any basic encoder with generating matrix G(D). Second, a general method, based on the Smith normal for matrices over a Euclidean ring is used to solve the syndrome equation for all possible error vectors e(D). The general solution of the syndrome equation for an (n, k) code is shown in (46) to have the explicit form,

$$e(D) = [\sigma_1(D), \ldots, \sigma_{n-k}(D), t_1(D), \ldots t_k(D)]\, L^{-1}(D)$$

where $t_i(D)$ for (i = 1, 2, ..., k) is an arbitrary polynomial in the

Euclidean ring F[D] and $\sigma_j(D)$ for ($j = 1, 2, \ldots, n - k$) computable linear functions of the $(n - k)$ syndromes $S_1(D), \ldots, S_{n-k}(D)$.

Next a Viterbi-like algorithm is developed to find the minimum Hamming-weight error-vector $\hat{e}(D)$ from all solutions of the syndrome equation. $\hat{e}(D)$ is the maximum likelihood estimate (MLE) of $e(D)$ within the coset of all solutions of the syndrome equation. The estimate $\hat{e}(D)$ is subtracted from the received codeword $z(D)$ to obtain the MLE $\hat{y}(D)$ of the transmitted codeword. Finally a method of Forney [7] and Massey and Sain [9] is extended and applied to find the inverse circuit needed to obtain the decoded message $\hat{x}(D)$ from $\hat{y}(D)$.

The new syndrome decoder for the (n, k) CC developed herein appears to be comparable in complexity to the Viterbi decoder. Possibly the control logic and the computations associated with the trellis are somewhat simpler in the new decoder. Another advantage of the new encoder may be obtained from the ease with which codes with the same constraint length, but with different rates, can be switched. Detailed comparisons of the new syndrome decoder with the Viterbi decoder are clearly topics for several future studies.

REFERENCES

1. I. S. Reed, *Introduction to the Use of Transforms for Encoding Reed-Solomon Codes,* First Quarterly Progress Report submitted to Naval Air Systems Command on Contract N00019-81-C-0541, March 1982.

2. I. S. Reed, *Further Results on Encoding Reed-Solomon Codes,* Second Quarterly Progress Report submitted to Naval Air Systems Command on Contract N00019-81-C-0541, June 1982.

3. E. R. Berlekamp, "Bit-Serial Reed-Solomon Encoders," *IEEE Trans. on Information Theory,* IT-28 (November 1982), 869-874.

4. I. S. Reed, *A General Symbol-Serial Multiplier for Reed-Solomon Codes*, Third Quarterly Progress Report submitted to Naval Air Systems Command on Contract N00019-81-C-0541, September 1982.

5. I. S. Reed and T. K. Truong, "Simplified Syndrome Decoding of (n, 1) Convolutional Codes," submitted to *IEEE Trans. on Communications*.

6. I. M. Herstein, *Topics in Algebra,* Xerox College Publishing, Lexington, Mass. (1975).

7. G. D. Forney, "Convolutional Codes: Algebraic Structure," *IEEE Trans. on Information Theory,* IT-16 (1970), 720-738.

8. A. A. Albert, *Modern Higher Algebra,* University of Chicago Press, Chicago, Ill. (1947).

9. J. L. Massey and M. K. Sain, "Inverses of Linear Circuits," *IEEE Trans. Comput.*, C-17 (1968), 330-337.

10. W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes*, 2nd ed., Cambridge, Mass., M.I.T. Press (1972).

11. R. J. McEliece, *The Theory of Information and Coding,* Reading, Mass.: Addison-Wesley Publishing Co. (1974).

12. A. Viterbi and J. Omura, *Digital Communication and Coding,* New York: McGraw-Hill Book Co. (1978).